Tony Barboza
Bobby Chisolm

Team 11

Part 1:

For part 1 we spent a lot of time trying to implement the TF-IDF rankings but could not get it to work. The transferring of the data and the rankings themselves cause us a lot of problems. We created the DocumentFreqTracker class to act as a “singleton” that would keep the data and process it so the indexer and searcher could use it but ran into problems with the data. We also implemented the document part of the TF-IDF’s but were unable to test it because we could not get the query part to work properly. The DocumentFreqTracker has 2 main hash maps, one is a (termFreqs) map of documentID to a hashmap of term’s mapped to their frequency in that document and the other is the map we used to calculate the TF-IDF score for the given document (documentVectors). We also were unable to run trec_eval because we could not get the code to work.

If we had all properly ranked run files we would run commands comparing the output :
./trec_eval -m Rprec -m map -m ndcg_cut.20
../src/main/java/data/test200/test200-train/train.pages.cbor-article.qrels
../src/main/java/output/DefaultRankingOutput.txt

With…
./trec_eval -m Rprec -m map -m ndcg_cut.20
../src/main/java/data/test200/test200-train/train.pages.cbor-article.qrels
../src/main/java/output/<EACH TF-IDF RANKING>

Then…
-Using the data comparing the ranking functions we would of determined questions like which variants of TF-IDF are best.

GUESS…
-All though we don't have data to back it up I would assume the regular Similarity performs better because Tf-idf because it is based on vector space model so it doesn't capture slight differences in words(punctuation and plural).

Part 2:

Bobby and I tried for hours and hours to try to understand how to build the ranking for each TF-IDF SMART notation. Because we did not produce the ranking files I can not use my SpearmanRank file to figure out which ranking is most comparable.

I did not have the rankings needed to compare all the Tf-idf vs the default Similarity.

My SpearmanRank works to my knowledge though and I tested it by comparing a standard Similarity vs a custom Similarity that we produced for last week's program.

OUTPUT:
I get the value of 0.6386 for spearman's rank
In addition im printing out the map of queries and corresponding spearman's values.